**19CSE301 - COMPUTER NETWORKS**

**Socket Programming**

**LAB-3 : (03-08-2021)**

* **R.Abhinav**
* **CB.EN.U4CSE19453**

1. **Implement Server program which receives a line in lower case from the client and modifies the line to upper case and sends client.**

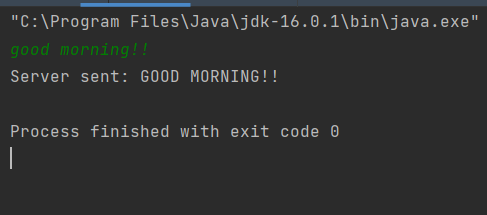
**Client :**

import java.io.\*;  
import java.net.\*;  
  
public class Client  
{  
 public static void main(String[] args)  
 {  
 String message = null;  
  
 try {  
 BufferedReader kbd =  
 new BufferedReader(new InputStreamReader(System.*in*));  
  
 Socket csock = new Socket("localhost", 6789);  
  
 DataOutputStream outNet =  
 new DataOutputStream(csock.getOutputStream());  
  
 BufferedReader inNet =  
 new BufferedReader(new InputStreamReader(csock.getInputStream()));  
  
 message = kbd.readLine();  
 outNet.writeBytes(message + "\n");  
  
 message = inNet.readLine();  
 csock.close();  
  
 System.*out*.println("Server sent: " + message);  
 }  
 catch(IOException e) {  
 System.*out*.println(e.getMessage());  
 }  
 }  
}

**Server :**

import java.io.\*;  
import java.net.\*;  
  
public class Server  
{  
 public static void main(String args[])  
 {  
 String messageIn;  
 String messageOut;  
  
 try {  
 ServerSocket ssock = new ServerSocket(6789);  
 while(true)  
 {  
 Socket connsock = ssock.accept();  
  
 InputStreamReader inStr =  
 new InputStreamReader(connsock.getInputStream());  
  
 BufferedReader inNet = new BufferedReader(inStr);  
  
 DataOutputStream outNet =  
 new DataOutputStream(connsock.getOutputStream());  
  
 messageIn = inNet.readLine();  
 messageOut = messageIn.toUpperCase() + "\n";  
 outNet.writeBytes(messageOut);  
 }  
 }  
 catch(IOException e) {  
 System.*out*.println(e.getMessage());  
 }  
 }  
}

**Output :**

****

1. **Implement Server program which receives a line from the client and count the vowels, number of characters and number of words and sends back to client**

**Server :**

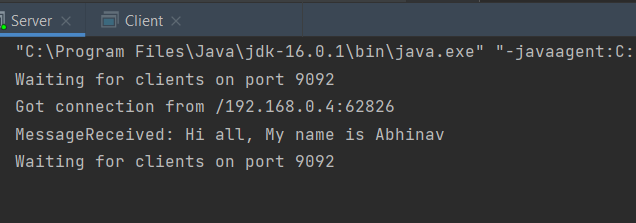
import java.io.BufferedReader;  
import java.io.InputStreamReader;  
import java.io.PrintWriter;  
import java.net.ServerSocket;  
import java.net.Socket;  
  
public class Server  
{  
 public static void main(String[] args)  
 {  
 int port = 9092;  
  
 try {  
 ServerSocket server = new ServerSocket(port);  
  
 while(true) {  
 System.*out*.println("Waiting for clients on port " + port);  
 Socket client = server.accept();  
  
 System.*out*.println("Got connection from "+client.getInetAddress()+":"+client.getPort());  
  
 BufferedReader reader = new BufferedReader(new InputStreamReader(client.getInputStream()));  
 PrintWriter writer = new PrintWriter(client.getOutputStream());  
  
 writer.println("Welcome to my server");  
 writer.flush();  
  
 String message = reader.readLine();  
  
 while (!(message == null || message.equalsIgnoreCase("exit"))) {  
 System.*out*.println("MessageReceived: "+message);  
 message = message.toLowerCase();  
 int a=0, b=0, c=0;  
 for (int i = 0; i<message.length(); i++)  
 {  
 if(message.charAt(i)=='a'||message.charAt(i)=='e'||message.charAt(i)=='i'||message.charAt(i)=='o'||message.charAt(i)=='u')  
 {  
 a = a+1;  
 }  
 if(message.charAt(i)==' ')  
 {  
 c = c+1;  
 }  
 }  
 {  
 writer.println("Vowels in message: " + a);  
 writer.println("Number of chars: "+message.length());  
 writer.println("Number of words: " + (c+1));  
 writer.flush();  
 }  
  
  
 message = reader.readLine();  
 }  
 client.close();  
 }  
 } catch(Exception ex) {  
 System.*out*.println("Connection error: "+ex);  
 }  
 }  
}

**Client :**

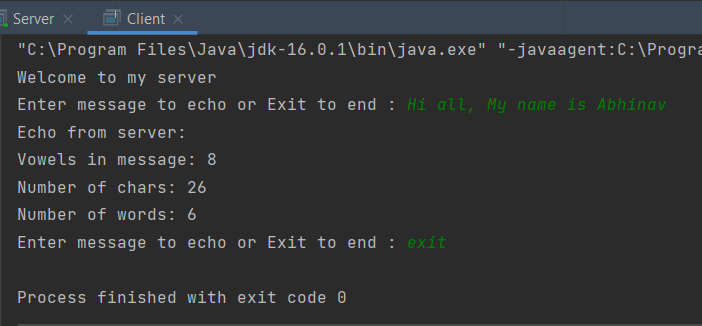
import java.io.BufferedReader;  
import java.io.InputStreamReader;  
import java.io.PrintWriter;  
import java.net.InetAddress;  
import java.net.Socket;  
  
public class Client  
{  
 public static void main(String[] args) {  
  
 int port = 9092;  
  
 try {  
 String host = InetAddress.*getLocalHost*().getHostName();  
 Socket client = new Socket(host, port);  
  
 PrintWriter writer = new PrintWriter(client.getOutputStream());  
 BufferedReader reader = new BufferedReader(new InputStreamReader(client.getInputStream()));  
  
 BufferedReader stdin = new BufferedReader(new InputStreamReader(System.*in*));  
  
 System.*out*.println(reader.readLine()); //read welcome message  
  
 String message;  
  
 while (true) {  
 System.*out*.print("Enter message to echo or Exit to end : ");  
 message = stdin.readLine();  
  
 if (message == null || message.equalsIgnoreCase("exit"))  
 break;  
  
 writer.println(message);  
 writer.flush();  
 System.*out*.println("Echo from server: \n"+reader.readLine());  
 System.*out*.println(reader.readLine());  
 System.*out*.println(reader.readLine());  
 }  
 client.close();  
  
 }catch (Exception ex) {  
 System.*out*.println("Exception: "+ex);  
 }  
 }  
}

**Output :**

**Server:**

****

**Client:**

****

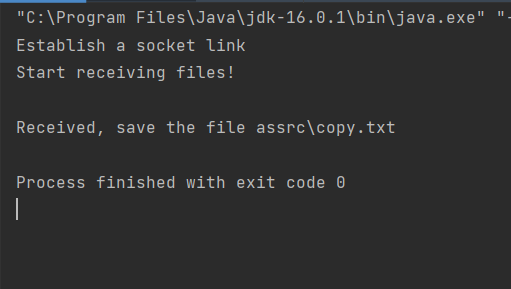
1. **Implement a text file transfer between server and client programs.**

* **Server :**
* import java.io.\*;  
  import java.net.ServerSocket;  
  import java.net.Socket;  
    
  public class Server {  
   private static final int *PORT*=1720;  
   public void downFile(String fileName) {  
   try {  
   ServerSocket serverSocket = new ServerSocket(*PORT*);  
   Socket socket = serverSocket.accept();  
   System.*out*.println("Establish a socket link");  
   DataInputStream inputStream = new DataInputStream(new BufferedInputStream(socket.getInputStream()));  
   byte[] buf = new byte[1024];  
   DataOutputStream fileOut = new DataOutputStream(new BufferedOutputStream(new FileOutputStream(fileName)));  
   System.*out*.println("Start receiving files!" + "\n");  
   while ((inputStream.read(buf))!=-1) {  
   fileOut.write(buf, 0,buf.length);  
   }  
   System.*out*.println("Received, save the file as" + fileName);  
   fileOut.close();  
   } catch (Exception e) {  
   System.*out*.println("Error receiving message");  
   e.printStackTrace();  
   return;  
   }  
   }  
    
   public static void main(String[] args) {  
   Server server=new Server();  
   server.downFile("src\\copy.txt");  
   }  
    
  }
* **Client :**

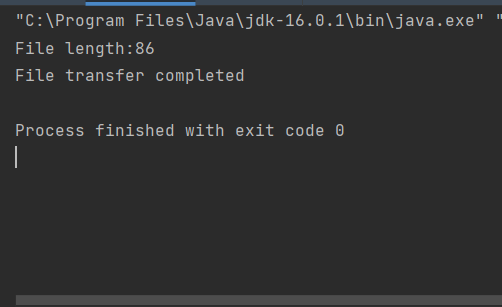
import java.io.\*;  
import java.net.Socket;  
  
  
public class Client {  
 // Define the address of the local port. Because it is on this machine, the port and address are defined as static constants  
 private static final String *HOST* = "localhost";  
 private static final int *PORT* = 1720;  
public void upFile(String fileName) {  
 try {  
 Socket socket = new Socket(*HOST*, *PORT*);  
 File file = new File(fileName);  
 System.*out*.println("File length:" + (int) file.length());  
 DataInputStream fileIn = new DataInputStream(new FileInputStream(fileName));  
 DataOutputStream out = new DataOutputStream(socket.getOutputStream());  
 byte[] buf = new byte[1024];  
 while ((fileIn.read(buf))!=-1) {  
 out.write(buf, 0, buf.length);  
 }  
 out.flush();  
 fileIn.close();  
 out.close();  
 socket.close();  
 System.*out*.println("File transfer completed");  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
   
 public static void main(String[] args) {  
 Client client = new Client();  
 client.upFile("C:\\first.txt");  
 }  
}

* **Output :**

**Server :**

****

**Client :**

****